Федеральное государственное автономное
образовательное учреждение
высшего образования
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт математики и фундаментальной информатики
Кафедра алгебры и математической логики

УТВЕРЖДАЮ
Заведующий кафедрой
_____ В. М. Левчук
«21» 06 2016г.

БАКАЛАВРСКАЯ РАБОТА

Направление 01.03.01 Математика

ПОСТРОЕНИЕ ТЕОРЕТИКО-ИНФОРМАЦИОННОЙ МОДЕЛИ СЛАБЫХ ПАРОЛЕЙ

Научный руководитель
Кандидат физико-математических
наук _____ / Ю. Ю. Ушаков
21.06.16

Выпускник _____ /И. С. Чжан
21.06.16

Красноярск 2016

РЕФЕРАТ

Выпускная квалификационная работа по теме «Построение теоретико-информационной модели слабых паролей» содержит 23 страниц текста, 2 использованных источника.

КРИПТОГРАФИЯ, ТОРЕТИКО-ИНФОРМАЦИОНАЯ МОДЕЛЬ, ПАРОЛЬ, КОНТЕКСТ, МОДЕЛЬ ЕСТЕСТВЕННОГО ЯЗЫКА, МОДЕЛЬ ОТКРЫТОГО ТЕКСТА.

Цель работы – научиться распознавать заведомо слабые пароли по модели, обученной естественному языку.

В результате была реализована программа для обучения и имитации модели по тексту естественного языка. Для вычисления условной вероятности программа обучается по тексту и считает вероятность встречи того или иного символа в контексте предыдущего. После чего получаем результат с подсчетами всех символов и контекстов.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Пароль – это кодовое слово или набор символов, предназначенных для подтверждения личности человека или его полномочий. Пароли часто используются для защиты информации от несанкционированного доступа. В большинстве вычислительных систем комбинация «имя пользователя — пароль» служит обеспечением контроля доступа к определённым ресурсам.

Есть два субъекта – пользователь, злоумышленник, и объект – информационная система (далее Система), которая хранит пароль.

Целью пользователя является получение доступа к его защищённой информации, которую хранит Система и предотвращение несанкционированного доступа. Целью злоумышленника является получение несанкционированного доступа к Системе, в том числе прочтение и модификация информации.

Подбор пароля является одним из способов получения несанкционированного доступа к Системе. Пароль – одна из составляющих безопасности Системы. Стойкость Системы определяется стойкостью самого слабого компонента Системы. При условии, что все остальные компоненты Системы являются стойкими, неграмотный выбор пользователями паролей становится критическим для Системы.

Пароли можно подбирать непосредственно в Системе, но такой подход часто ограничивается средствами Системы, например, количеством запросов с одного IP адреса и временем ожидания ответа. Другим видом атаки является подбор пароля по некоторой известной информации, например, по хэш-свертке пароля.

Существуют разные виды атак на пароли, и стойкость пароля должна определяться относительно каждого вида атак, так как пароль, стойкий к одному виду атак, может оказаться нестойким к другому виду атак.

Классическими атаками являются: полный перебор символов (или «метод грубой силы», от англ. brute force), подбор пароля по словарю. Но

возможны и другие атаки, например, атака с использованием дополнительной информации о теоретико-информационных характеристиках пароля. Многие пароли имеют вид «*НаступилоЛето!*». В таком пароле прослеживается много закономерностей, наследуемых от естественного языка, т. е. если пароль состоит из фрагментов естественного языка (слов или фраз), то эти закономерности переносятся на пароль. Поэтому атака может быть проведена с учетом этих закономерностей.

В работе исследуется возможная атака, основанная на закономерностях естественного языка. Была написана программа на языке программирования C++, которая строит модель пароля, основанную на закономерностях естественного языка и осуществляет проверку соответствия вводимого пароля модели естественного языка, тем самым определяя стойкость пароля к такой атаке.

1 Теоретико-информационная модель Шеннона

Основная идея теоретико-информационной модели естественного языка, заключается в учете вероятностей появления тех или иных символов независимо, либо в зависимости от предыдущих символов.

Обозначим через вероятность того, что следующая буква может идти с той или иной вероятностью, в зависимости от - последовательностей предыдущих букв.

Пусть представляющий собой массив, состоящий из приближений для вероятностей – появления -грамм в отрытом тексте, алфавит открытого текста, , . Тогда источник *открытого текста* генерирует последовательность знаков алфавита в которой -грамма появляется с вероятностью , следующая *-грамма* появляется с вероятностью и т. д. Назовём построенную модель открытого текста вероятностной моделью -го приближения.

Таким образом, простейшая модель открытого текста – вероятностная модель первого приближения – представляет собой последовательность знаков в которой каждый знак появляется с вероятностью , независимо от других знаков. Будем называть также эту модель *позначной моделью открытого текста.* В такой модели открытый текст имеет вероятность

.

В вероятностной модели второго приближения первый знак имеет вероятность , а каждый следующий знак зависит от предыдущего и появляется с вероятностью

,

( ) ( ) ( ) ( ) ( )

где Другими словами, модель открытого текста второго приближения представляет собой простую однородную цепь Маркова. В такой модель открытого текста $c_1c_2…c_l$ имеет вероятность ( ) ( ) П (|

.

Модели открытого текста более высоких приближений учитывают зависимость каждого знака от большего числа предыдущих знаков. Ясно, что чем выше степень приближения, тем более «читаемыми» являются соответствующие модели.

Модели естественного языка используются для проверки некоторой последовательности символов, чтобы определить, является ли она последовательностью на естественном языке (соответствует ли она модели). Так же модель может быть использована для имитации, например, в нашем случае по модели слабого пароля генерируются варианты для проверки их Системой. Но перед использованием все вероятности (|
должны быть оценены, процесс их оценки называется обучением модели и производится по подборке текстов.

1.1 Алгоритм обучения

Обучение модели происходит по некоторой выборке символов, по которой оценивается вероятность встречи того или иного символа. Стоит заметить, что подборка текстов для обучения модели должна быть репрезентативной, т. е. посчитанные по ней оценки должны соответствовать теоретической вероятности. На практике это обозначает, что используемые для обучения тексты должны быть близки по стилю и тематике к текстам, которые предполагается исследовать.

В нашей задаче при подборе пароля можно использовать дополнительную информацию о жертве. Например, представители интернет-

субкультуры вероятнее всего, будут использовать принятые в их среде аббревиатуры и сокращения, вроде «втф», «лол» и т. д. С другой стороны, при оценке качества пароля, где дополнительная информация не известна, следует использовать для обучения модели комбинацию самых различных текстов, в том числе и субкультурные «наречия», т. е. подобрать всевозможные тексты.

Введём понятия *окно* и *контекст*. *Окном* будет являться последовательность символов. Вероятность появления последнего символа в окне зависит от предыдущих символов, назовём их *контекстом*.

Для обучения модели последовательности символовперемещаем окно по тексту, мы проходим все *-граммы* символов и считаем вероятность встречаемости того или иного символа

(|

.

Пример алгоритма обучения. Построим по фразе модели нулевого, первого и второго порядков. Для примера возьмем пятибуквенный алфавит {а, в, и, н, _} и последовательность символов «иван_нива_вина». Символ «_» в данном случае является пробелом.

Модель нулевого порядка:

Всего символов 14, из них «и» встречается 3 раза. Поэтому

/

.

Аналогично находим остальные вероятности.

/ ;

/ ;

/ ;

.

Модель первого порядка:

Символ «и» является контекстом 3 раза, из них «в» следует после «и» в двух случаях, а «н» в одном случае. Отсюда, , .

Символ «в» является контекстом 3 раза, из них «а» следует после «в» в двух случаях, а «и» в одном случае. Отсюда, , .

Символ «а» является контекстом 2 раза, из них «н» следует после «а» в одном случае, «_» следует после «а» в одном случае (в одном случае «а» является концом последовательности символов). Отсюда, , .

Символ «н» является контекстом 3 раза, из них «_» следует после «н» в одном случае, «и» следует после «н» в одном случае и «а» следует после «н» в одном случае. Отсюда, , , .

Символ «_» является контекстом 2 раза, из них «н» следует после «_» в одном случае и «в» в одном случае. Отсюда, , .

Модель второго порядка:

«ив» является контекстом 2 раза, из них «а» следует после «ив» в обоих случаях. Отсюда, .

«ва» является контекстом 2 раза, из них «н» следует после «ва» в одном случае и «_» в одном случае. Отсюда, , .

«ва» является контекстом 2 раза, из них «н» следует после «ва» в одном случае и «_» в одном случае. Отсюда, , .

«ан» является контекстом 1 раза, после «ан» следует «_». Отсюда, .

(| «н_)» является контекстом 1 раза, после «н_» следует «н». Отсюда,

.

Аналогично вычисляются остальные вероятности.

1.2 Алгоритм имитации

После обучения программы по тексту естественного языка, производим имитацию модели.

Считаем, что модели от 0-го до *n*-го порядка уже построены. Приведем пример имитации, используя модель, построенную в предыдущем пункте.

Первую букву берём случайным образом по модели нулевого порядка.

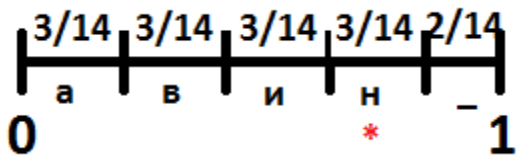

Рисунок 1 – Имитация модели нулевого порядка

Разделим отрезок от 0 до 1 на интервалы между этими буквами в соответствии с оценками их вероятностей, как показано на рисунке 1. Выбираем случайное число, используя равномерное распределение от 0 до 1 и выпишем букву, в интервал которой попало случайное число. Наше случайное число попало в интервал буквы «н».

Следующая буква имитируется на основе модели нулевого порядка, используя предыдущую сымитированную букву. Теперь на отрезке от 0 до 1 откладываем уже условные вероятности (рисунок 2). (| /

; ; .

Рисунок 2 – Имитация модели первого порядка

Наше случайное число попало в интервал «ни».

Аналогично генерируются последующие буквы. Например, мы сгенерировали фразу «нива_», следующая буква генерируется с помощью модели максимального имеющегося порядка, т.е. в нашем случае порядка 2. Поскольку в модели вероятность , то следующей буквой будет «в» и т. д.

1.3 Алгоритм проверки на соответствие модели

Построив модель текста, мы теперь можем сравнивать имеющиеся тексты с этой моделью, т.е. выполнять проверку соответствия текста модели. При этом можно воспользоваться либо стандартными методами различения статистических гипотез, либо наличием в открытых текстах некоторых запретов, таких, например, как биграмма ЪЪ в русском тексте. Проиллюстрируем первый подход при распознавании позначной модели открытого текста.

Итак, согласно нашей договоренности, открытый текст представляет собой реализацию независимых испытаний случайной величины, значениями которой являются буквы алфавита появляющиеся в соответствии с распределением вероятностей

Требуется определить, является ли случайная последовательность букв алфавита открытым текстом или нет.

Пусть – гипотеза, состоящая в том, что данная последовательность – открытый текст, – альтернативная гипотеза. В простейшем случае последовательность можно рассматривать при гипотезе как случайную и равновероятную. Эта альтернатива отвечает субъективному представлению о том, что при расшифровании криптограммы с помощью ложного ключа получается "бессмысленная" последовательность знаков. В более общем случае можно считать, что при гипотезе последовательность представляет собой реализацию независимых испытаний некоторой случайной величины, значениями которой являются буквы алфавита , появляющиеся в соответствии с распределением вероятностей При таких договоренностях можно применить, например, наиболее мощный критерий различения двух простых гипотез, ко-торый дает лемма Неймана-Пирсона.

В силу своего вероятностного характера такой критерий может совершать ошибки двух родов. Критерий может при-нять открытый текст за случайный набор знаков. Такая ошибка обычно называется ошибкой первого рода, ее вероятность равна . Аналогично вводится ошибка второго рода и ее вероятность . Эти ошибки определяют качество работы критерия. В криптографических исследованиях естественно минимизировать вероятность ошибки первого рода, чтобы не "пропустить" открытый текст. Лемма Неймана-Пирсона при заданной вероятности первого рода минимизирует также вероятность ошибки второго рода.

Критерии на открытый текст, использующие запретные сочетания знаков, например, *k-граммы* подряд идущих букв, будем называть критериями запретных *k-грамм*. Они устроены чрезвычайно просто. Отбирается некоторое число *s* редких *k-грамм*, которые объявляются запретными. Теперь, просматривая последовательно *k-грамму* за *k-граммой* анализируемой последовательности мы объявляем ее случайной, как только в ней встретится одна из запретных *k-грамм*, и открытым текстом в противном

случае. Такие критерии также могут совершать ошибки в принятии решения. В простейших случаях их можно рассчитать. Несмотря на свою простоту, критерии запретных *k-грамм* являются весьма эффективными.

На основе модели, построенной в предыдущем параграфе, разберем соответствие модели на примере двух слов *ива* и *ванна*.

Буква «и» появляется с вероятностью

Вероятность встретить «в» после «и» равна

Вероятность встретить «а» после «ив» равна

Как можно заметить, слово *ванна* не соответствует нашей модели, так как два последних множителя

2 Возможные математические модели слабого пароля

Слабый пароль, как правило, состоит из нескольких лексем. Например, пароль «Лето2016» состоит из двух лексем, буквенной «Лето» и цифровой «2016». Мы рассматриваем отдельно буквенные лексемы, отдельно цифровые лексемы, и отдельно рассматриваем модель, которая определяет в каком порядке идут эти лексемы.

Объединяющая модель строится на основе двухбуквенного алфавита {а, б}, где «а» – буквенная часть, «б» - цифровая часть. Эта модель также состоит из вероятностей $P(\cdot|\cdot)$, где, например, $P(б|аа)$ обозначает вероятность того, что после цифровой и буквенной лексем следует снова буквенная лексема.

Например, последовательность «ЛетоЗима2016» состоит из двух буквенных и одной цифровой лексемы. Поэтому в объединяющей модели ей соответствует слово «ааб».

2.1 Обучение модели

Для обучения буквенной модели используются тексты на естественном языке, для обучения цифровой модели используется список дат, объединяющая модель обучается по списку паролей.

Символы каждого пароля рассматриваются по порядку, когда следующий символ идет из другого подмножества по сравнению с предыдущим, тогда это означает, что началась новая лексема, если же очередной символ из того-же подмножества, что и предыдущий, тогда осуществляется проверка по модели. Если вероятность следующей буквы в контексте предыдущих меньше заданного порога, то началось новое слово.

Например, «ЛетоЗима2016». Как мы видим после слова «Лето» следует буква «З». Буква «З» в контексте «Лето» встречается с вероятностью примерно равной нулю. Это означает, что после слова «Лето» начинается новое слово с буквы «З», т. е. «Зима». Далее проходим до конца слова «Зима» и определяем, что дальше следует цифра, значит началась новая лексема, причем цифровая. Следовательно, наша обобщающая модель будет представлена в виде «ааб».

Обучение по словарю паролей происходит следующим образом: каждый пароль разбивается на лексемы, буквенные лексемы заменяются на «а», цифровые – на «б»; объединяющая модель обучается по всем таким словам.

2.2 Алгоритм проверки на соответствие модели

Вначале разбиваем пароль на лексемы в алфавите {а, б}, после находим вероятность каждой лексемы. Когда вероятности каждой лексемы найдены, находим вероятность объединяющей лексемы. После чего можно посчитать вероятность кодовой фразы.

2.3 Примеры проверки паролей

Для наглядного примера возьмем две возможные парольные фразы: «Hello2016» и «gfgoqerac».

Проверим оба этих пароля на ресурсах, где есть проверка сложности пароля. Были выбраны следующие системы: «Яндекс», «Мэйл.ру», и программа

для хранения и генерации паролей «KeePass». Затем проверим эти же пароли на нашей программе.

Как видно на рисунке 3, «Мэйл.ру» считает пароль «Hello2016» не очень надежным. Но если мы попробуем ввести тот же самый пароль два раза, т.е. «Hello2016Hello2016», то эвристика системы считает, что пароль соответствует всем требованиям и является надежным (рисунок 4).

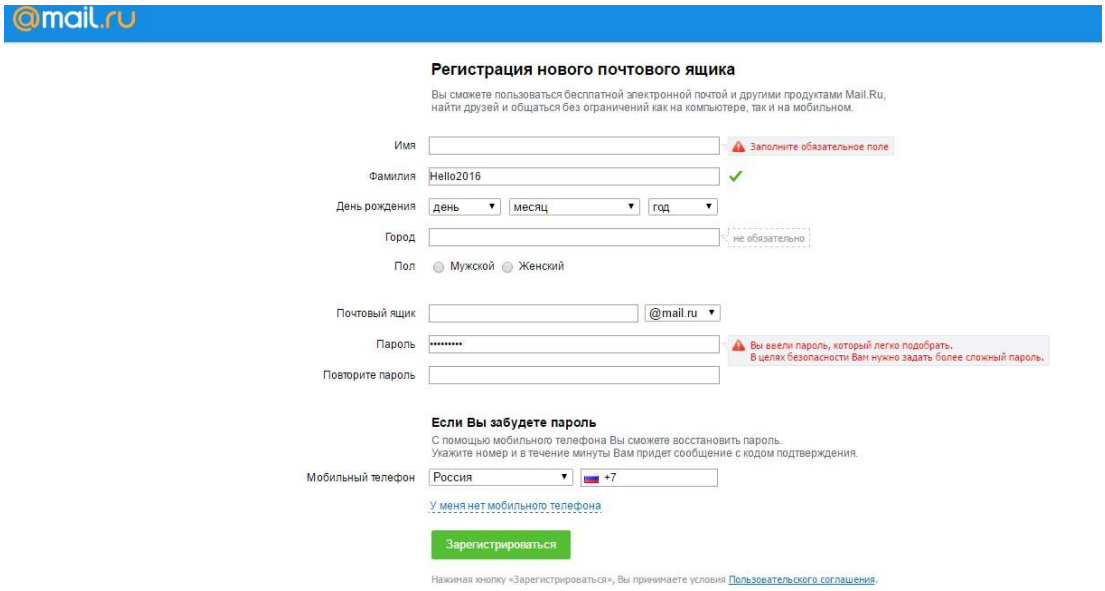

Рисунок 3 – Проверка пароля «Hello2016» в «Мэйл.ру»

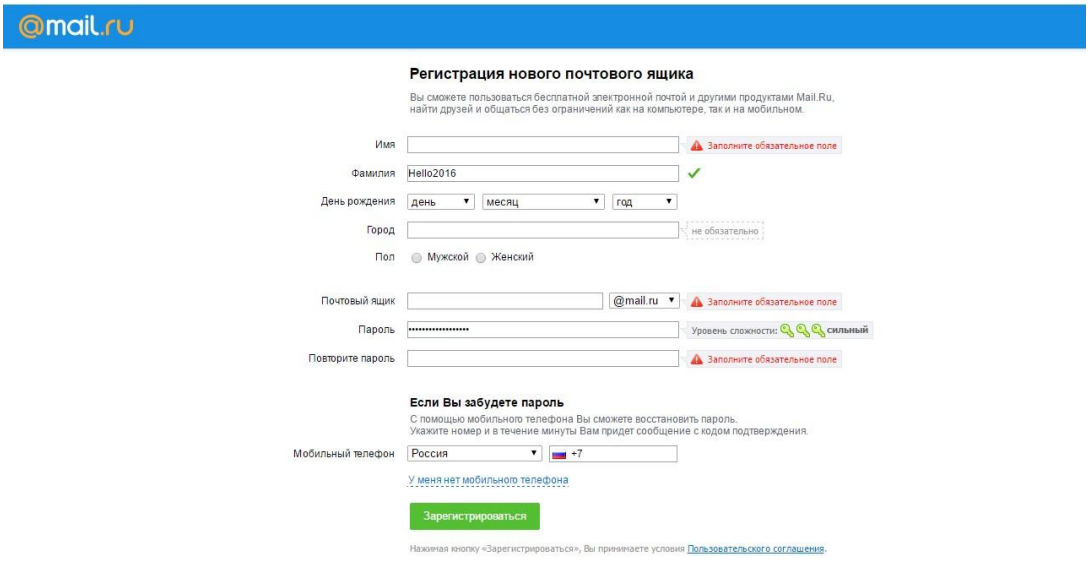

Рисунок 4 – Проверка пароля «Hello2016Hello2016» в «Мэйл.ру»

Система «Яндекса» сразу признала пароль ««Hello2016» надежным (рисунок 5). Скорее всего это связано с тем, что используются буквы разного регистра и цифры. Пароль «Hello2016Hello2016» стал еще надежнее. Увеличив количество символов в 2 раза, «Яндекс» сообщает нам, что всё хорошо (рисунок 6).

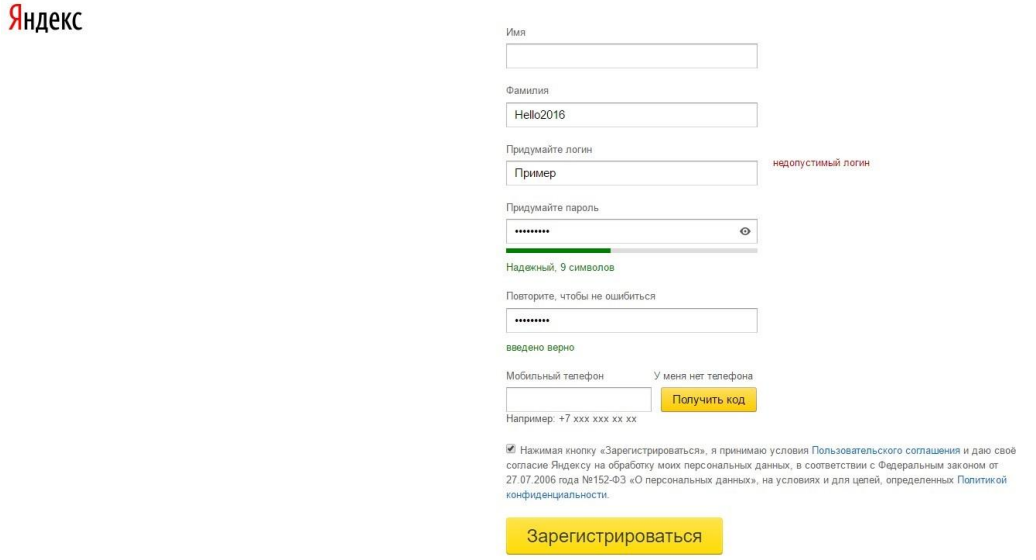

Рисунок 5 – Проверка пароля «Hello2016» в «Яндекс»

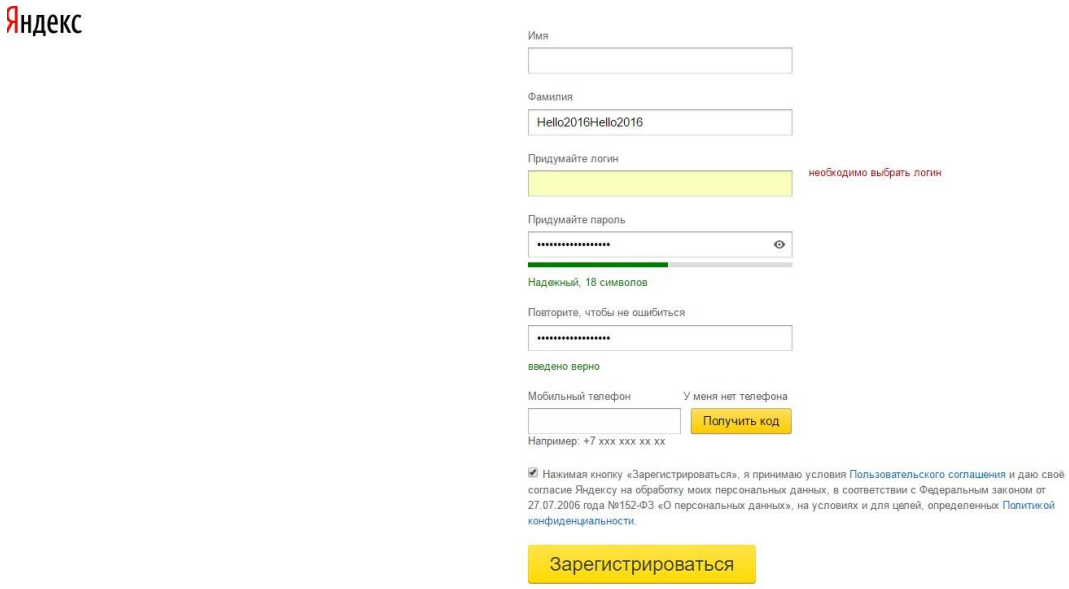

Рисунок 6 – Проверка пароля «Hello2016Hello2016» в «Яндекс»

Программа «KeePass», созданная для работы с паролями (создание, хранение, защита), определила, что пароль «Hello2016» не является надежным (рисунок 7). Даже увеличение длины за счет удвоения пароля не дало особо заметных результатов. Пароль по-прежнему не является надежным, несмотря на длину (рисунок 8).

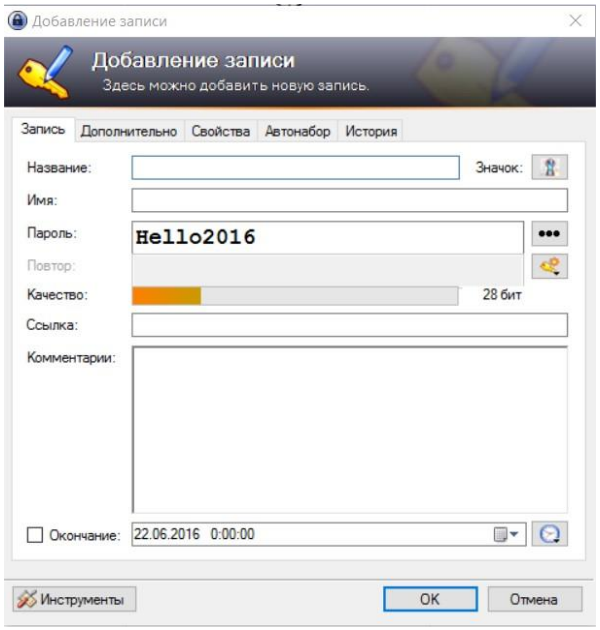

Рисунок 7 – Проверка пароля «Hello2016» в программе «KeePass»

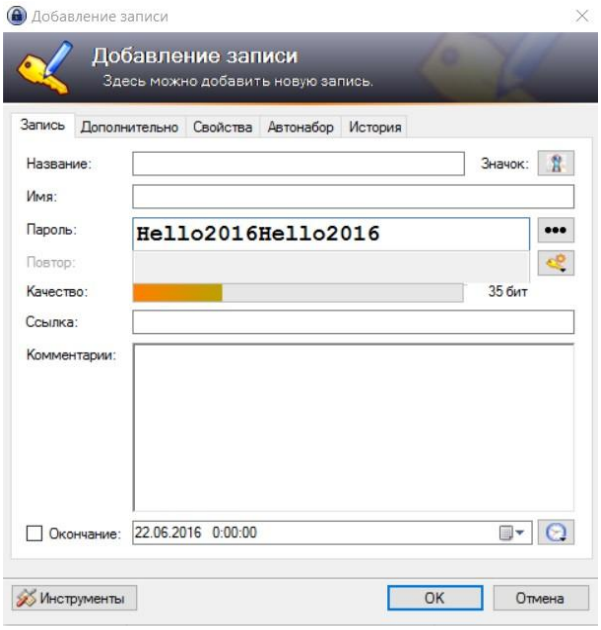

Рисунок 8 – Проверка пароля «Hello2016Hello2016» в программе «KeePass»

Теперь рассмотрим случай с паролем «gfgoqerac».

«Мэйл.ру» по-прежнему говорит, что пароль, который указываем при регистрации не является надежным (рисунок 9), а при вводе пароля «gfgoqeracgfgoqerac» показывает стойкость 2/3 (рисунок 10). Что вполне может соответствовать действительности.

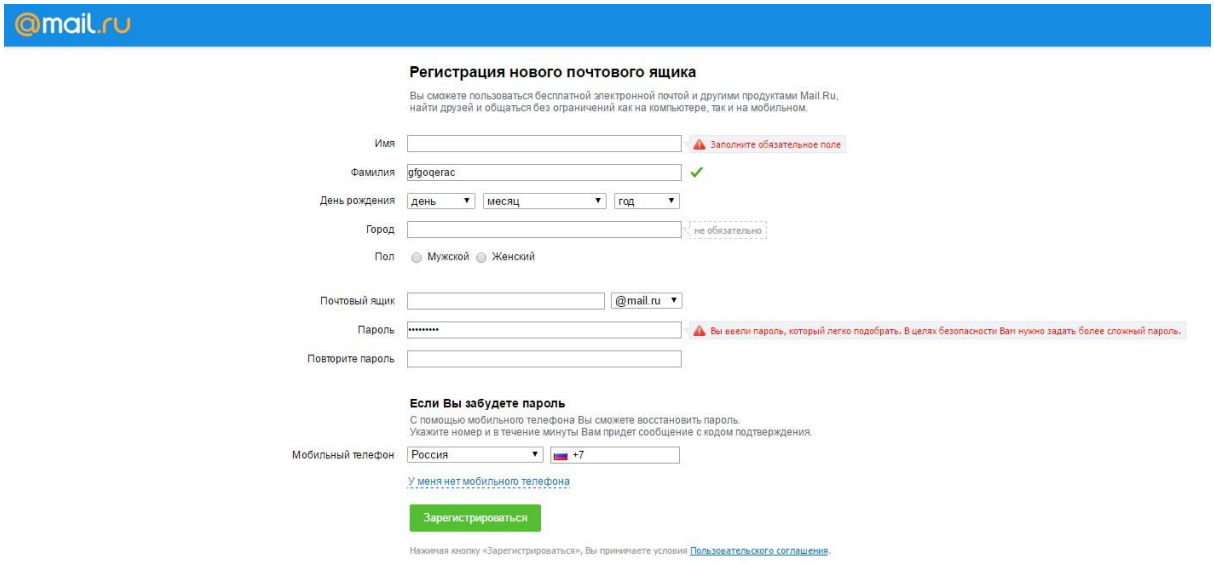

Рисунок 9 – Проверка пароля «gfgoqerac» в «Мэйл.ру»

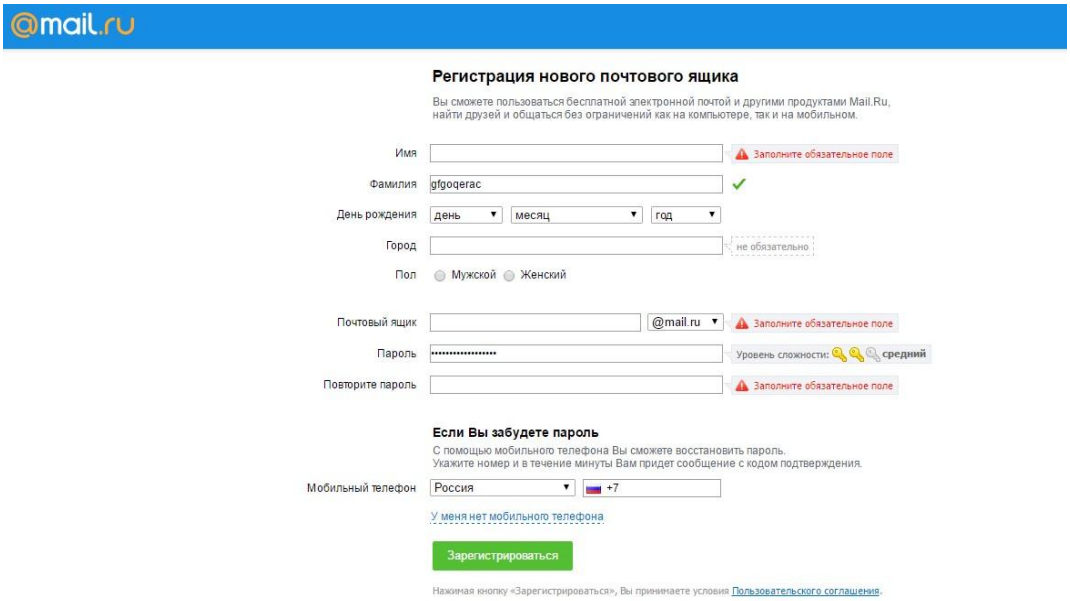

Рисунок 10 – Проверка пароля «gfgoqeracgfgoqerac» в «Мэйл.ру»

«Яндекс», как и в случае с паролем «Hello2016», позывает такое же значение устойчивости (рисунок 11 и 12).

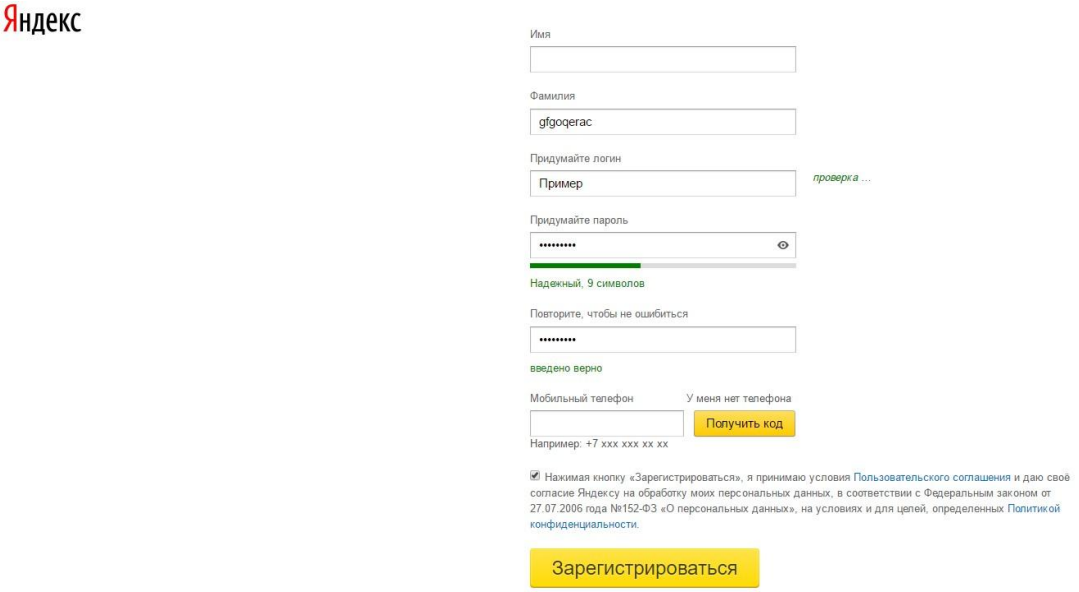

Рисунок 11 – Проверка пароля «gfgoqera» в «Яндекс»

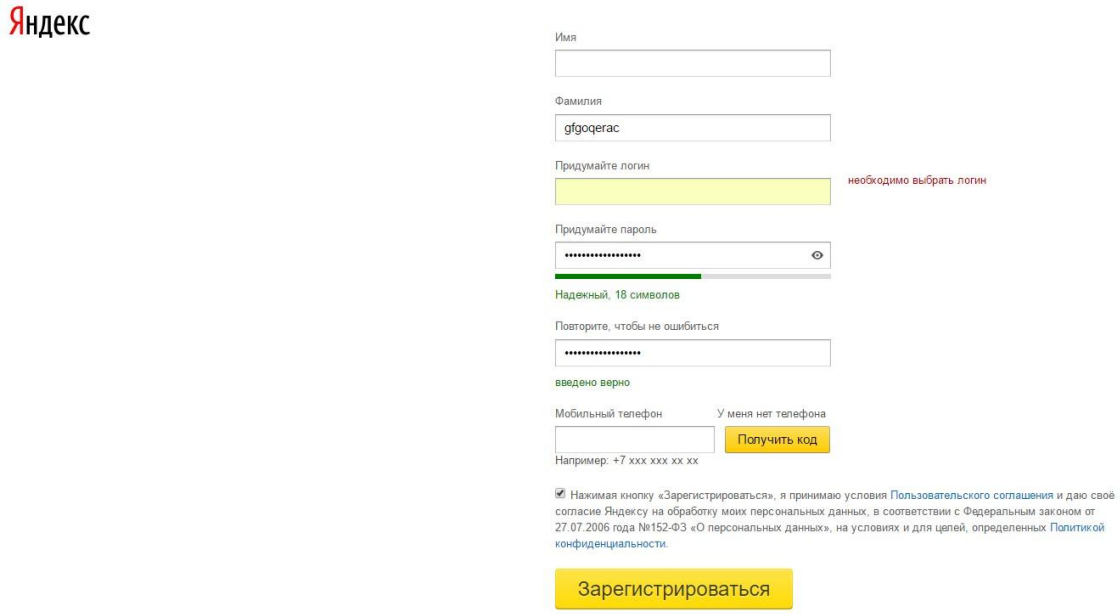

Рисунок 12 – Проверка пароля «gfgoqeragfgoqera» в «Яндекс»

«KeePass» справляется с задачей оценки пароля. Используя специальные алгоритмы, достаточно достоверно отражает *качество* вводимого пароля (рисунок 13 и 14).

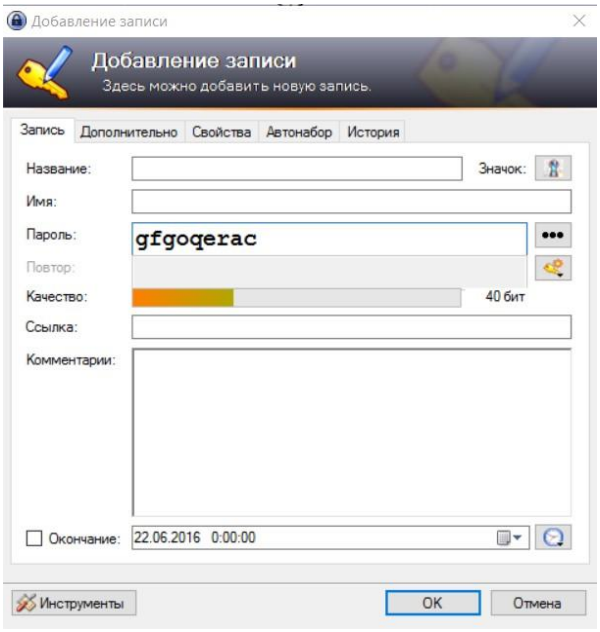

Рисунок 13 – Проверка пароля «gfgoqera» в программе «KeePass»

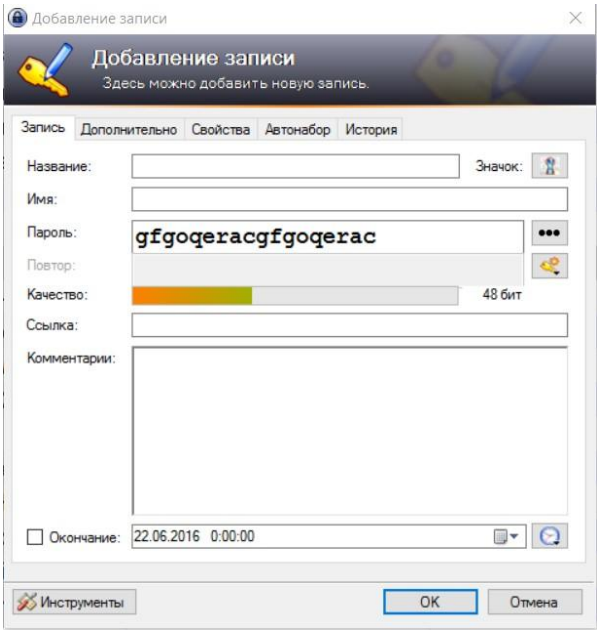

Рисунок 14 – Проверка пароля «gfgoqeragfgoqera» в программе «KeePass»

Проверив данные пароли на нашей программе, получили следующие результаты: ( ) ( ) (| ||) (|) (|

(|

( ) ( ) (| , (| |) (|

(|

.

Таким образом можно заметить, что пароль более стойкий к атаке полным перебором является не стойким для перебора по модели.

ЗАКЛЮЧЕНИЕ

В ходе исследования, мы убедились, что стойкость пароля должна определятся отдельно относительно каждого типа атак, т. к. пароль стойкий к перебору «грубой силой» или по словарю оказывается не стойким к перебору по модели, обученной естественному языку и умеющей разбивать слова на лексемы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Алферов, А.П. Основы криптографии / А.П. Алферов, А.Ю. Зубов и др. — Москва: Гелиос АРВ, 2002, – 480 с.

2 Шеннон, К. Работы по теории информации и кибернетике / Шеннон К. — Москва: Издательство иностранной литературы, 1968, – 832 с.